then the MRPI of the MRPF is read in step 218. Then in step 219, the method tests if the PRI's hook field is in an MRPI. It will rarely be true because this means the PRI was then referenced before it was created. If so, the PRI pointer is added to the MRPI in step 220 and then method proceeds to step 221. If the PRI's hook field is not a MRPI, the method proceeds directly to step 221 where the method determine if this is the last MRPI to process. If there are additional MRPIs, the method returns to step 218 otherwise the method continues to step 216. In step 216, the method tests whether there are additional MRTs to process. If there are, the method jumps to step 212. If not, the method test for additional PRI to process in step 21. For any more PRI, the method returns to step 211, or else the process is completed.

Referring now to FIGS. 11A, 11B, and 11C, the preferred method for updating a MRT for a changed PRI is described. Similar to the method of FIG. 10, the process begins by reading the changed PRI in step 230, reading the PRT definition and the where used list of this PRI in step 231, and reading the definition of the MRT listed in where used list in step 232. The method next tests whether this a lead PRT for a MRPF in step 234. If the PRT is a lead PRT for a MRPF, the method continues with step 244 of FIG. 11C. In step 244, the process determines if a key field was changed. If a key field was changed, the method executes step 245, 246 and 247, to respectively find the old MRPI, remove the old MRPI, and create a new MRPI by executing the new PRI routine of FIG. 17A for this MRT. However, if a key field was not changed, the method finds the old MRPI in step 248, changes the MRPI single occurrence pointer for the changed hook fields in step 249, and then runs the functions and filter for the changed PRIs and MRPIs. After competition of either branch, the method returns to step 241 of FIG. 11B.

Referring back to 11A, if the PRT is not a lead PRT for a MRPF, the method continues in step 235 by reading the lead PRIs for the MRPF. Then in step 236, the method compares the old hook fields values to determine if they match. If there is a match, the PRI is changed by removing the pointer in the MRPI in step 237 before proceeding to step 238. In step 238, the method tests if the new hook values match. If there is no match the method jumps to step 241. If there is a match, the PRI is changed by creating new pointer in the MRPI in step 239. Then the method runs the functions and filters for the changed PRIs and MRPIs in step 240. Then the method test whether the current process involves the last PRT, the last MRT or the last changed PRI in steps 241, 242, and

243, respectively. If this is not the last PRT of the last MRT for the last changed PRI, then the method loops to the appropriate step (i.e., steps 235, 231 and 230) for further processing.
The preferred method for executing a function for a changed PRI is shown in FIGS. 12A and 12B. The present invention handles PRT fields that are used in a group by function or used in the function of a child MRPF especially for recalculation purposes. Whenever one of these PRT's fields change, present invention not only recalculates the functions for its own MRPIs but also recalculates the group by function value and the other functions in which it is used. In other words, present invention treats these fields as calculated fields by finding where they are used and making sure that the additional appropriate recalculations take place. The present invention does this by keeping pointers to the group by functions and functions of the other MRPFs that use the field.

The preferred method begins by reading the PRI and the PRT definition in steps 260 and 261, respectively. Then in step 262, the function using the PRI is read. Next in step 263, the MRT's MRPF for the function is read. Whenever a PRT's PRI changes, the present invention finds the MRPFs that use that PRT and tests whether any of its functions use the changed fields. In step 264, the method determines if the PRT is a lead PRT of the MRPF or used in a group by function. If so, the method continues in step 269 by reading the MRPI to which the PRI points. In each MRPI, the present invention stores the MRPI for each MRPF parent. Each parent MRPI that has group by functions using any of the changed fields has its group by value recalculated. Recalculation does not require reprocessing of all the records that are included in the group. The present invention stores enough information about each group by field to recalculate the correct balance for the before and after image of the changed field. For example, for the average field, present invention also stores the number of values used as well as the actual average. New averages are calculated by multiplying the number of values times the average then subtracting the old value and adding the new value and then dividing by the stored number of values. Then the function is executed in step 270. There functions are executed in a conventional manner. The process is similar to that used in spreadsheets and third generation languages. In step 271, the functions for any changed MRPI are executed, after which the method jumps to step 268. However, if the PRT is not a lead PRT of a MRPF or in a group by function, the method proceeds to step 265 where the MRPI is read. Next, in step 266, the method tests whether the MRPI points to a PRI. Next, the present invention checks to see it any of the calculated fields that the present invention just updated are used in other group by functions. If

so, it recalculates the group by value in the appropriate parent MRPF's MRPI. The present invention continues this loop of cascading group by executions until the set of calculated fields that are recalculated have no impact on any other function because those group by fields are not used in any other function. If it does point to a PRI, the function is executed in step 272, and the functions for any changed MRPI are executed in step 273. The method then continues in step 267 or arrived at step 267 directly from step 266. In step 267, the method tests whether there are any other MRPI for the changed PRI. If there are additional MRPIs, the method returns to step 265 to read another MRPI and process it as has been described above. If there are no more MRPIs, the method moves to step 268 and tests whether this is the last function for the modified PRI. If not, the method returns to step 262 to retrieve and process the next function. Once all the functions have been processed the present method is complete.

Generally functions are executed whenever the user accepts some changes made to a pane row. The user has the option of deferring the execution of the affected functions until he/she wishes. Deferring changes can speed up entry of changes. The present invention knows which other functions to run because whenever a function is defined or maintained, the present invention adds a pointer to the Where Used Pointers of the PRT Fields Definition and the Where Used Pointers of the Function Definition.

FIG. 13 illustrates the preferred method for maintaining a function and its corresponding calculated fields in MRPIs and DDPs. The process begins by accepting user changes to the function in step 280. Then the method replaces the calculated fields used in the function with its function. Next, in step 282 the method updates the PRTs used in the function to point to the function. Then in step 283, the MRPF or the DDP targeted to point to the function are updated. In step 284 the MRPI, pane row for the MRPI, or the DDP is read. Finally, the function is run in step 285. Then the method tests for other MRPI or PRTs affected by the function change and returns to step 284 to process them. Once all the MRPIs and PRT have been processed, this method ends.

FIG. 14 illustrates the preferred method for executing a filter if the filter has been modified. The method starts by accepting the user's changes to the filter in step 290. Then in step 291, the method updates the MRT definition or DD 20 pane definition to point to the filter. Next, the method